AMENDMENT TO CLAIMS

All pending claims are reproduced below.

1. (Previously presented) A computer-implemented system to process an XML document, comprising:

a streaming parser operable to parse an XML document to generate a stream of events, wherein each event in the stream represents a portion of the document;

a matching component operable to:

accepting the stream of events from the streaming parser;

keeping in memory only a subset of the stream of events at any time;

performing a match on an event in the subset of the stream of events; and

notifying an observer if the event is a matched event;

said observer operable to listen for the matched event and passing it to a user object; and

said user object operable to handle the matched event.

2. (Previously presented) The system according to claim 1, wherein:

the XML document is represented in a hierarchical structure.

3. (Previously presented) The system according to claim 2, wherein:

the hierarchical structure is a tree with each node containing a portion of the document.

4. (Previously presented) The system according to claim 3, wherein:

the streaming parser is operable to generate the stream of events by:

traversing the XML tree and adding visited nodes into a data structure;

processing the nodes in the data structure and generating an event for each node; and

appending the event to the output stream.

5. (Previously presented) The system according to claim 4, wherein:

the tree is traversed using a breath-first or depth-first search.

6. (Previously pesented) The system according to claim 4, wherein:

the data structure is a queue.

7. (Previously presented) The system according to claim 4, wherein:

the data structure is processed using a first-in-first-out approach.

8. (Previously presented) The system according to claim 1, wherein:

the matching component is operable to keep only a portion of the XML document in memory at any given time.

9. (Previously presented) The system according to claim 1, wherein:

the matching component is operable to know the schema of the XML document and foreseeing the coming events.

10. (Previously presented) The system according to claim 1, wherein:

the match is an expression-based match, which can be an XPath query.

11. (Previously presented) The system according to claim 3, wherein:

the matching component is operable to keep, clone and destroy the entirety or a portion of the sub-tree descending from a node in the tree.

12. (Previously presented) The system according to claim 1, wherein:

the user object is operable to return the matched event to an XML stream for use by any other component.

13. (Previously presented) A method for processing an XML document, comprising:

parsing an XML document to generate a stream of events, wherein each event in the stream represents a portion of the document;

accepting the stream of events and keeping in memory only a subset of the stream of events at any time;

performing a match on an event in the subset of the stream of events;

notifying an observer if the event is a matched event;

listening for the matched event and passing it to a user object; and

handling the matched event.

14. (Previously presented) The method according to claim 13, further comprising:

representing the XML document in a hierarchical structure, which is a tree with each node containing a portion of the document.

15. (Original) The method according to claim 14, wherein:

the parsing of the XML document comprises the steps of:

traversing the XML tree and adding visited nodes into a data structure;

processing the nodes in the data structure and generating an event for each node; and

appending the event to the output stream.

16. (Original) The method according to claim 15, wherein:

the XML tree is traversed using a breath-first or depth-first search.

17. (Original) The method according to claim 15, wherein:

the data structure is processed using a first-in-first-out approach.

18. (Original) The method according to claim 13, further comprising:

keeping only a portion of the XML document in memory at any given time.

19. (Original) The method according to claim 13, further comprising:

knowing the schema of the XML document and foreseeing the coming events.

20. (Previously presented) The method according to claim 13, further comprising:

performing an expression-based match, which is an XPath query.

21. (Original) The method according to claim 14, further comprising:

keeping, cloning and destroying the entirety or a portion of the sub-tree descending from a node in the tree.

22. (Previously presented) The method according to claim 13, further comprising:

returning the matched event to an XML stream for use by any other component.

23. (Previously presented) A machine readable medium having instructions stored thereon that when executed by a processor cause a system to:

parse an XML document to generate a stream of events, wherein each event in the stream represents a portion of the document;

accepting the stream of events and keeping in memory only a subset of the stream of events at any time;

perform a match on an event in the subset of the stream of events;

notify an observer if the event is a matched event;

listen for the matched event and pass it to a user object; and

handle the matched event.

24. (Original) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

represent the XML document in a hierarchical structure, which ~~can be~~ is a tree with each node containing a portion of the document.

25. (Original) The machine readable medium of claim 24, further comprising instructions that when executed cause the system to:

parse the XML document, comprising the steps of:

traversing the XML tree and adding visited nodes into a data structure;

processing the nodes in the data structure and generating an event for each node; and

appending the event to the output stream.

26. (Original) The machine readable medium of claim 25, further comprising instructions that when executed cause the system to:

traverse the tree using a breath-first or depth-first search.

27. (Original) The machine readable medium of claim 25, further comprising instructions that when executed cause the system to:

process the data structure using a first-in-first-out approach.

28. (Previously presented) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

perform an expression-based match, which is an XPath query.

29. (Original) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

keep only a portion of the XML document in memory at any given time.

30. (Original) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

know the schema of the XML document and foresee the coming events.

31. (Original) The machine readable medium of claim 24, further comprising instructions that when executed cause the system to:

keep, clone and destroy the entirety or a portion of the sub-tree descending from a node in the tree.

32. (Previously presented) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

return the matched event to an XML stream for use by any other component.

33. (Previously presented) A system for processing an XML document, comprising:

means for parsing an XML document to generate a stream of events, wherein each event in the stream represents a portion of the document;

means for accepting the stream of events and keeping in memory only a subset of the stream of events at any time;

means for performing a match on an event in the subset of the stream of events;

means for notifying an observer if the event is a matched event;

means for listening for the matched event and passing it to a user object; and

means for handling the matched event.

34. (Canceled).